第3章 カー・エレクトロニクスにおけるプロセッサの役割とは

——車線維持システムのアルゴリズムとプロセッサ選び

鈴木伸夫

ここでは，セーフティ系システムの一つである，走行車線の維持を支援する「車線維持システム」を取り上げ，そのアルゴリズムと実装について解説する．車線維持システムは，カメラから取り込んだ画像をもとに車線の位置を検出し，自動車が車線から逸脱しそうな場合にドライバに警告するシステムである．この一連の処理のほとんどがソフトウェアによって実現されている．（編集部）

さまざまな分野に普及してきたプロセッサ（マイクロコントローラ）は，自動車業界にもその勢力を伸ばしています．なかでも「アクティブ・セーフティ」と呼ばれる自動車の安全分野では，高い処理能力が要求されるため，高性能なプロセッサが導入されています．本稿では，「車線維持システム」を例にとって，アクティブ・セーフティにおけるプロセッサの役割と機能を説明します．

1 安全性の追求とプロセッサへの要求

今日の自動車技術は，エレクトロニクスによって支えられています．かつてはすべてが機械式制御であった自動車も，今や電子化されていない機能は一つもないと言えるほどです．この数十年の間に，環境制御やエンジン制御，エンターテインメントなど，エレクトロニクスを基盤とした最新技術が提供され，さらなる進化を続けています．この進化の原動力となった要因の一つが，プロセッサ技術の進歩です．

自動車業界におけるプロセッサの歴史は，1970年代の排気ガス規制のために採用されたエンジン制御用のマイクロコントローラに始まります．その後，さまざまな種類の車載用プロセッサが開発され，エンジンやその他の制御機構を中心に用途が広がりました．最近では，カー・ナビゲーション・システムに代表されるマルチメディア機器や，「環境」，「安全」，「快適」をテーマとした最新車載機器においても，プロセッサが重要な役割を果たしています．

とくに，高機能化が著しいセーフティ系システムは，各自動車メーカが新技術の開発にしのぎを削っており，プロセッサに対する要求も高度になってきています．自動車のいろいろな場所に設置されているセンサから送られてくる信号だけでなく，リア・カメラや前方監視カメラなどのビデオ映像を取り扱うケースも多く，膨大なデータを瞬時に解析できる高い演算能力のプロセッサが求められます．そのため，信号処理機能を強化したマイクロプロセッサやディジタル信号処理プロセッサ（DSP：digital signal processor）など，マルチメディア処理に優れたプロセッサが導入されています（pp.80-81のコラム「DSPの基礎知識」を参照）．

● パッシブ・セーフティとアクティブ・セーフティ

自動車の安全対策技術は，大きく二つに分けることができます．一つは「パッシブ・セーフティ」と呼ばれるもので，シートベルトやエア・バッグなど，避けられない事故に遭遇したときに被害を最小限に抑えるための技術です．車体の設計をくふうし，衝突時の衝撃を適切に分散・吸収することによってキャビン・スペースの衝撃を緩和する「衝突安全設計ボディ」は，自車の乗員だけでなく歩行者や対向車両の被害をも軽減するようになっています．

もう一つの安全技術は，危険を予知して回避することによって事故を未然に防ぐ「アクティブ・セーフティ」と呼ばれるものです．滑りやすい路面での急ブレーキに対してタイヤがロックされるのを防止する「ABS（Antilock Brake System）」は，アクティブ・セーフティのさきがけです．ア

クティブ・セーフティは，エレクトロニクスの発展とともにさらなる進化を続けており，車速を制御し先行車との車間を適切に保持する「車間距離制御システム」や走行車線の維持を支援する「車線維持システム」，横滑りなどの不安定な車体を自動的に補正する「姿勢制御システム」など，今日もさまざまな新しいシステムが開発されています．

● 車線維持システムのほとんどの処理はプロセッサで実行

アクティブ・セーフティの代表的なシステムとして，「車線維持システム」が挙げられます．車線維持システムとは，危険な車線変更を試みたときや，走行中の車線から車体が逸脱しそうになったときに，ドライバに注意を促すシステムです(図1)．前方監視カメラによって白線や縁石に対する車体の相対位置を監視し，車体が車線から逸脱しそうになると警報を発します．さらに，パワー・ステアリング・システムを制御して，車体が車線の中央付近を走行するようにドライバの操作を補助する場合もあります．

車線維持システムにおける処理の大部分は，ビデオ・カメラの映像から車線を認識するための画像解析です．図2に，システムの概要を示します．話を簡単にするため，ここではパワー・ステアリング・システムの制御は行わず，車線の逸脱に対して警告を発するだけのシステムを想定します．この場合，画像の解析処理の多くはプロセッサとソフトウェアによって実現することができます．周辺のアナログ回路などの役割はごくわずかです．このことは，システム全体で使われる部品点数の削減につながり，省スペースや省コストなどのさまざまなメリットをもたらします．

プロセッサは，カメラ・ユニットから入力される道路の実画像に対して，ノイズ除去などの前処理を施した後，道路表面の輪郭だけを抽出し，ドライバが維持するべき走行車線の境界を割り出します．この処理を繰り返し行うことによって走行中の車線をつねに監視し，走行位置の乱れを検出すると，ドライバに通知するかどうかの判定を行います．

2 車線維持システムにおけるプロセッサの役割

次に，車線維持システムの基本的な構成要素を探ってみましょう．図3は図2と同じ処理フローを示していますが，実行される処理のアルゴリズムに沿った流れになっています．

● 画像入力ではサイズと転送速度のトレードオフを考慮

自動車のセーフティ系アプリケーションにおける画像解像度は，QVGA(320×240ピクセル)からVGA(640×480ピクセル)程度のサイズがよく用いられます．実際には，画像サイズは一つのトレードオフですが，転送データ量が少なければ転送速度が下がり，プロセッサの負担が軽くなります．車線維持システムでは，グレイ・スケールの画像で十分です．そのため，クロマ(色相)情報を無視することができ，データ帯域幅を画素16ビット/ピクセルから8ビット/ピクセルへ減らすことができます．

通常，カメラ・ユニットから入力される画像データは，一時的にSDRAMのような大容量の外部メモリに格納された後，プロセッサが内蔵する高速メモリに順次取り込まれ

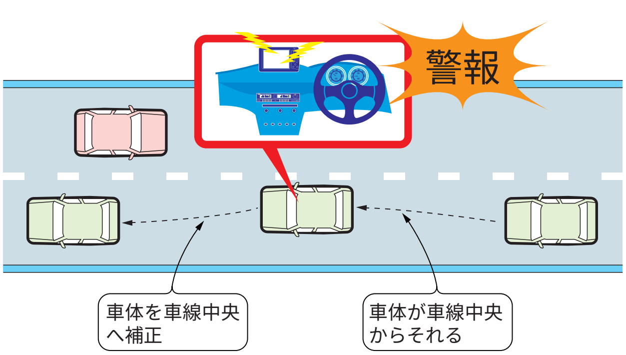


図1 車線維持システム

車線維持システムは，危険な車線変更を試みたときや居眠り運転などで，走行中の車線から車体が逸脱しそうになった場合に警報を鳴らしてドライバに注意を促す．

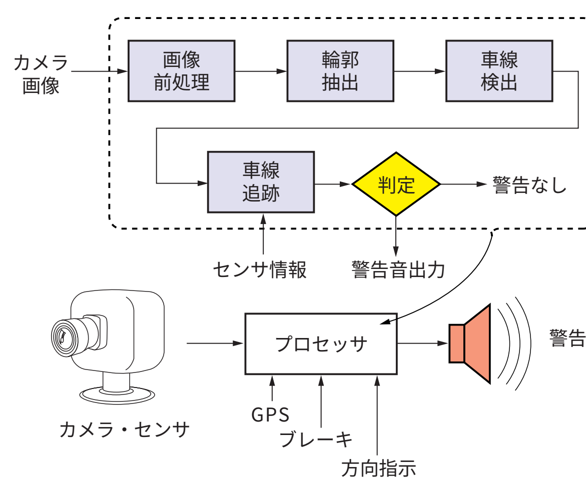


図2 車線維持システムの基本構成

車線維持システムは，カメラ，プロセッサ，スピーカなどで構成されている．ほとんどの処理はプロセッサ上でソフトウェア処理される．

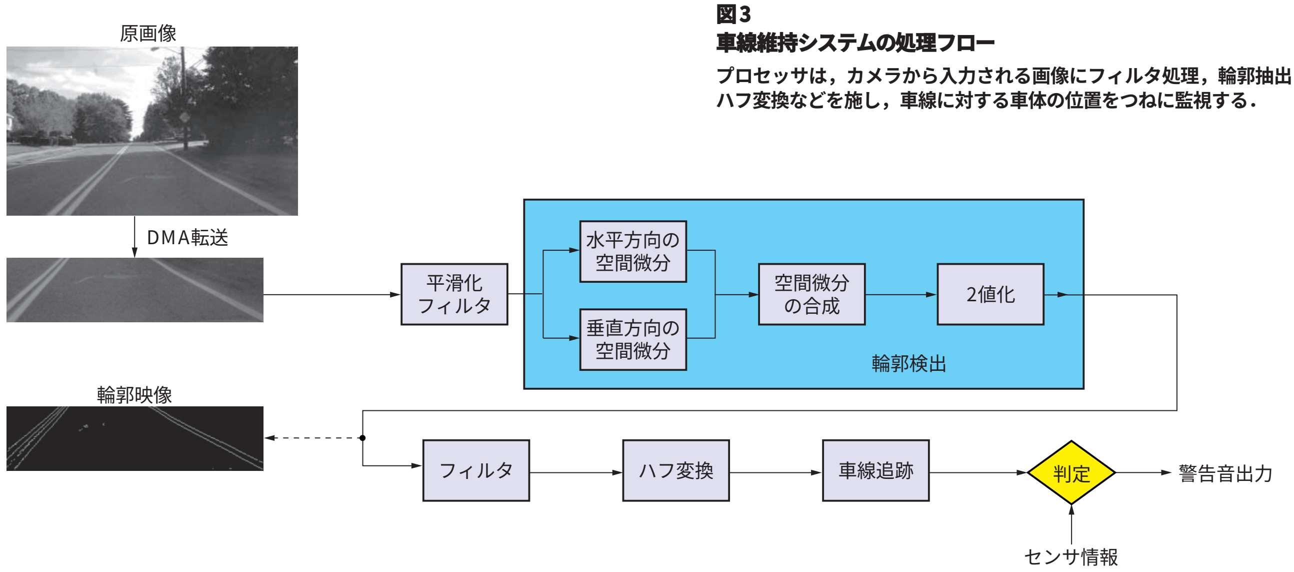


図3
車線維持システムの処理フロー
プロセッサは，カメラから入力される画像にフィルタ処理，輪郭抽出，ハフ変換などを施し，車線に対する車体の位置をつねに監視する．

ます(**図4**)．メモリ間で行うデータ転送には，一般にDMA(direct memory access)を用います．例えばDSPの場合，DMAを利用すると，内部に取り込まれたデータをプロセッサが演算処理すると同時に，次に必要なデータを取り込むことができます．その結果，絶えず送られてくる画像データを取りこぼすことなく，適切な演算処理を継続して行うことができます．

● フィルタリングには積和演算の得意なDSPを利用

システムに入力されたビデオ・ストリームは，温度変化や車体の振動，電磁波の干渉などの影響により，さまざまなノイズ成分を含んでいます．ノイズ成分を含む画像では車線を正確に検出できないので，ノイズ成分を除去するフィルタリングと平滑化処理を行います．このとき，プロセッサの処理速度は画像の入力速度に追従できなければなりません．そのため，ソフトウェアをできる限り少ない処理サイクルで実行するように最適化します．

画像フィルタの場合，中核となる2次元畳み込み演算を中心に最適化します．2次元の畳み込みにおける一つの画素に対する演算は，その画素に隣接する画素の重み付き総和演算です．隣接画素の抜き出しには，一般的に3×3マスクが使用されます．3×3マスクの構成例を**図5**に示します．例えば，画像内の10行20列目の画素に対する畳み込み演算は，以下のようになります．

$$
\begin{aligned}
\mathrm{Out}(20,10) &= A \times (19,9) + B \times (19,10) + C \times (19,11) \\
&\quad + D \times (20,9) + E \times (20,10) + F \times (20,11) \\
&\quad + G \times (21,9) + H \times (21,10) + I \times (21,11)
\end{aligned}
$$

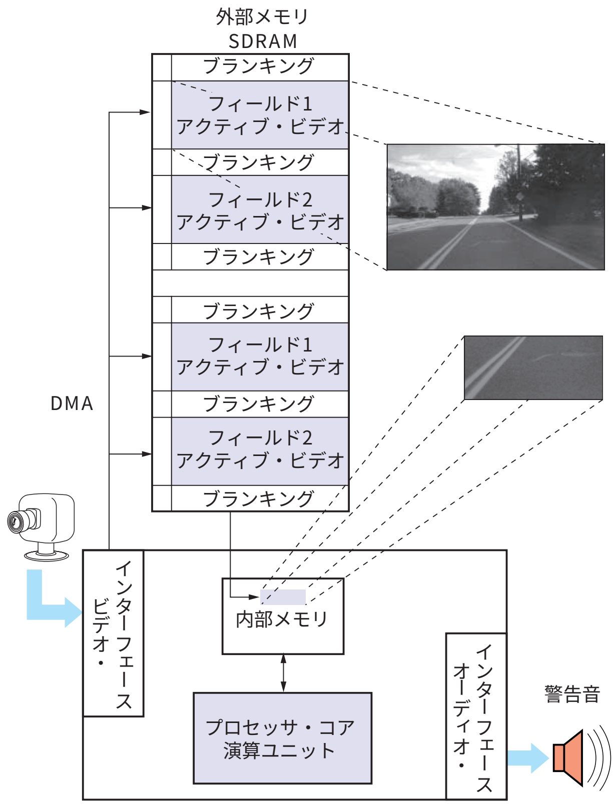


図4　外部メモリを利用したフレーム・バッファの例
ビデオ・インターフェース経由でカメラから取り込まれる画像は，一時的に外部メモリに蓄えられる．プロセッサは，外部メモリ上の画像から必要な部分だけを内部メモリに取り込み，演算処理を行う．

この演算は，乗算と加算を対とした積和演算の繰り返しになります．積和演算といえばDSPがもっとも得意とする

図5
3×3マスクによる畳み込み演算
畳み込み演算では，1個の出力値に対して9個の乗算と8個の加算演算が必要である．これをすべての画素に対して繰り返し行う．なお，A～Iはマスク係数を表す．

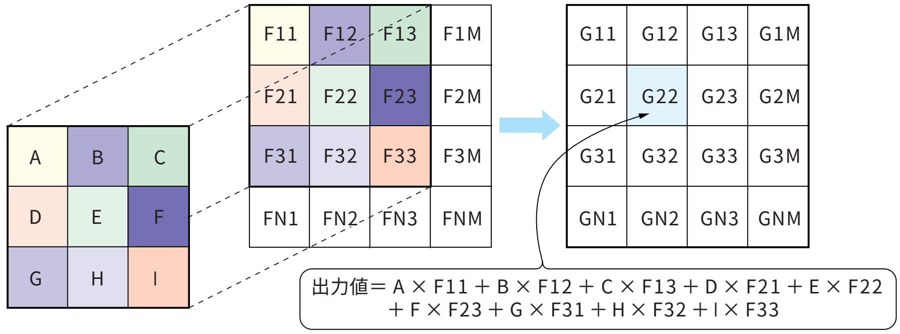
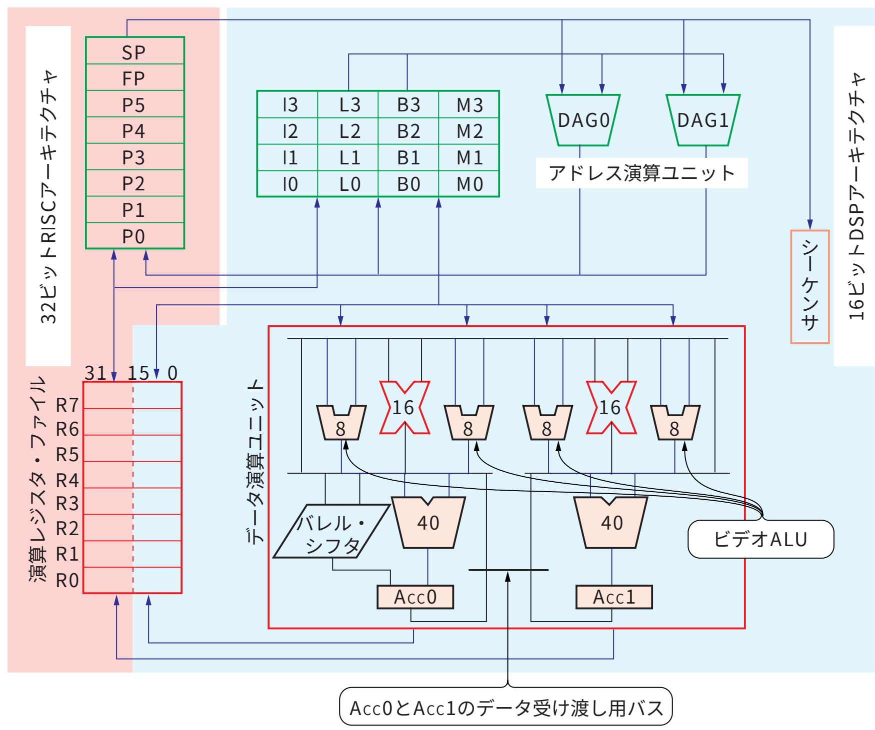


図6
アルゴリズムに合ったプロセッサを選ぶ
図は，Analog Devices社のプロセッサ「Blackfin」のコア・アーキテクチャ．二つの積和演算ユニットや四つのビデオALUなどのDSP的なアーキテクチャに加え，SP（スタック・ポインタ）/FP（フレーム・ポインタ）などのRISC的なアーキテクチャも採用している．このほか，パラレル・ペリフェラル・インターフェース（PPI）として，車載機器で一般的に使用されているITU-R656準拠のCMOSカメラやLCDモニタに直接接続することができる．また，PPIと連動するDMAコントローラは2次元転送に対応している．



演算の一つです．例えば，米国Analog Devices社のプロセッサ「Blackfin」は二つの積和演算ユニットを内蔵しています（**図6**）．1サイクルで二つの積和演算を行えるので，2次元の畳み込み演算を高速に処理できます．また，RISCプロセッサ並みの制御処理性能を備えており，ステアリング制御や警告支持などにも対応できます．

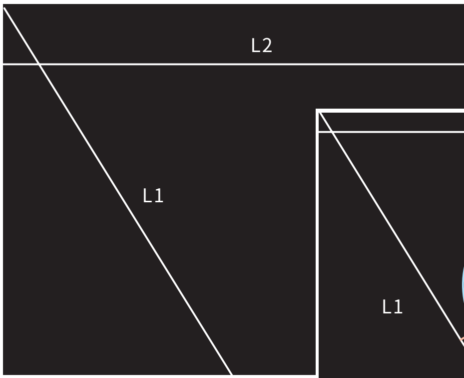
ソフトウェアの最適化も重要ですが，使用するアルゴリズムに適したプロセッサを選択することもシステムを設計する際の重要な要素の一つです．

● 二つの畳み込み演算回路を使って輪郭抽出

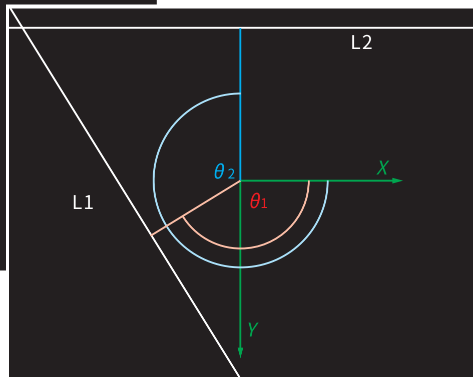
一般的な輪郭（エッジ）抽出法では，複数の微分演算子の組み合わせを利用します．各微分演算子は，画像内の濃淡が変化する場所を検出できるように設計されます．ここでは，ソベル・フィルタによる輪郭抽出法を紹介します．ソベル・フィルタは，二つの畳み込みカーネルを使用して，水平方向と垂直方向の輪郭に対する空間微分を計算します．以下に，垂直方向の濃淡の変化を検出する演算子S_x，水平方向の濃淡の変化を検出する演算子S_yを示します．

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

出力行列の各要素は，画像の各画素が輪郭である可能性の高さを示しています．ここで，画像に含まれる輪郭部分の出力値が大きくなることを利用して，出力行列を2値化



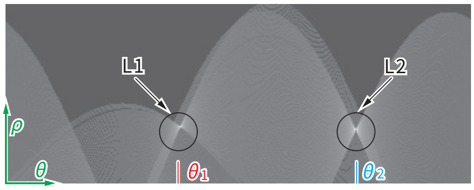
(a) 輪郭抽出により，2値化された画像



(b) ハフ変換におけるパラメータ表示

図7
ハフ変換のイメージ

(a)は前段の輪郭抽出で得られる画像が2値化されており，"直線"を視覚的に見ることができる．この例では2本の直線が検出されている．(b)は，検出された2本の直線が，各直線に対する原点からの垂線(青と赤の直線)の長さと角度で表記できることを示す．(c)では，θの範囲は[0,2π]，ρの範囲は(a)の入力画像の対角線の1/2．図中の明るい2点が(a)の2本の直線に対応する．



(c) ハフ変換の結果

します．その結果得られた濃淡のない白黒画像が，後段のハフ変換(Hough transform)への入力となります．

● ハフ変換を利用した車線の割り出し

ハフ変換は，画像内の線や円，く形(長方形)などの抽象的な特徴を抽出する場合に広く用いられています．式(1)に示す極表現を基盤とするハフ変換空間上では，画像上の「線」を「点」として表せることから，ハフ変換は車線検出に適した手法といえます．

$$\rho = x\cos\theta + y\sin\theta \qquad (1)$$

式(1)の意味は，与えられた線における原点を通る垂線を用いて図示することができます．θはX軸と垂線のなす角を表し，ρは垂線の長さを意味します．したがって，1組の(ρ, θ)によって直線を表現できます．**図7**にこの概念を示します．**図7(a)**の2本の線L1とL2は，**図7(b)**ではそれぞれθ_1と赤い垂線の長さ，θ_2と青い垂線の長さで表現できます．

このハフ変換を適用した車線検出のアルゴリズムは，次のようになります．

1) 2値画像の白画素だけに着目
2) 各画素のθに注目し，原点から角度θの画素に対して直線を引く
3) 加算テーブルにこの(ρ, θ)を登録
4) 上記1)～3)の処理を，画像内のすべての白画素に対して繰り返す
5) 加算テーブルを検索し，もっとも頻繁に出現する(ρ, θ)の組を見つける

ここで得られた(ρ, θ)が道路の白線の目印を含む可能性がもっとも高い直線の組み合わせです(**図7(c)**)．さらに，カメラの中心軸からのオフセットや検出された直線の幅，カメラの位置に対する角度など，さまざまな付加情報を利用して，対象の車線と車体の位置を割り出します．

● 再帰フィルタを使って車線追跡

ハフ変換で各画像から抽出される車線情報は，連続する複数の画像間で相関性があります．この相関性を求めることで，車線維持システムは車体が車線を逸脱していないかどうかの判定を行います．この相関性を追跡するために，一般に利用されているのがカルマン・フィルタ(Kalman filter)です．

カルマン・フィルタは，物体の未来の状態を推定する再帰フィルタとして知られています．車線維持システムにおける物体は車線であり，車線の状態はその位置と複数の画像フレームにわたる移動経路で表現されます．また，カルマン・フィルタを用いると，道路の状態そのものに加えて，その状態に対する分散も得られます．予測された状態とその分散は，次の画像フレームにおけるハフ変換の検索領域を狭めるために使用することができ，結果的に処理時間を短縮することができます．

● 危険判定では誤判定を防ぐくふうが必要

プロセッサの最後のしごとは，ドライバへの通知です．どのようなタイミングで，どのような方法でドライバへ通

3

COLUMN DSPの基礎知識

車載用途のプロセッサの多くは，システム制御やデータ処理を得意とするマイクロコントローラやマイクロプロセッサです．しかし，カー・ナビゲーション・システムや本文で述べたセーフティ系システムなどにマルチメディア機能が導入され始め，DSPを使用する機会も多くなっています．DSPは，ディジタル信号処理を高速に行うことを目的に開発されたプロセッサです．ここでは，DSPを語るうえで欠かせないいくつかの基礎用語を紹介します．

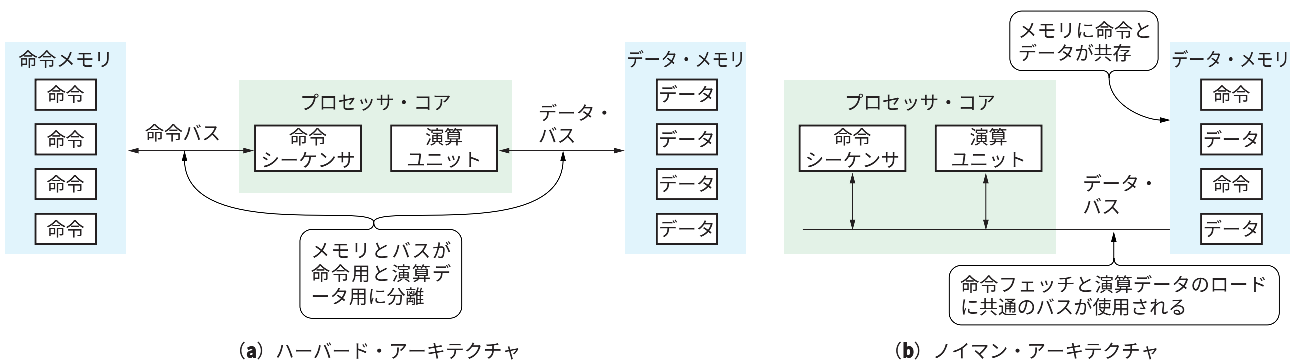
● ハーバード・アーキテクチャ

メモリに保存されている命令とデータをプロセッサ・コアが読み込む(フェッチする)ために，命令用とデータ用で異なるバスが用意されているアーキテクチャを「ハーバード・アーキテクチャ」と呼びます(図A-1(a))．DSPは，ハーバード・アーキテクチャを採用しており，命令のフェッチと演算データの読み込みを同時に行うことができます．そのため，メモリ・アクセスが連続する演算も効率良く処理することができます．

一方，マイクロプロセッサなどで採用されているノイマン・アーキテクチャでは，命令とデータの読み込みに同じバスを使用するため，命令の読み込み中にデータを読み込むことができません(図A-1(b))．そのため，コアの性能がバス・アクセスの性能によって制限されてしまいます．

● 積和演算(MAC)ユニット

本文の畳み込み演算のところでも紹介しましたが，ディジタル信号処理のアルゴリズムの多くは，乗算の結果を加算する累積加算演算を多用します．この累積加算を「積和演算」または「MAC(multiply and accumulate)演算」と呼び，DSPは積和演算を1サイクルで行うための演算ユニットを持っています．この演算ユニットを「積和演算ユニ



図A-1　ハーバード・アーキテクチャとノイマン・アーキテクチャ
(a)のハーバード・アーキテクチャでは，メモリとバスが命令用と演算データ用で分離しており，それぞれが独立に動作する．一方，(b)のノイマン・アーキテクチャは，命令フェッチと演算データのロードに共通のバスを使用するため，それぞれの動作がバス上で互いに競合する．

知するかを選択するのもプロセッサの役割です．しかし，ここで注意しなければならないのは，誤判定(false positive)です．

セーフティ系システムが実際にはない危険を誤って検出してしまった場合，それに対する応答が実行されるよりも早く，ユーザがそのシステムの機能を停止させる方法はありません．そのため，システムの設計者は危険判定の処理段階で誤った判断がなされないような機能を機器に組み込む必要があります．

例えば，もっとも簡単な例として，方向指示器(ウィンカ)情報を利用する方法があります．たとえ車線維持システムが車線の逸脱を検出したとしても，方向指示器が作動していれば(つまり，ドライバが自分の意志で車線変更を行っていれば)，それは正常な状態と言えます．このようにして，誤判定を避けることができます．より高度なシステムでは，GPSの情報や過去の運転履歴，日時，天候といったさまざまな情報を利用して，誤判定を防いでいます．

＊　＊　＊

ここでは，プロセッサを利用した車載機器の例として，車線維持システムの構成とそこに実装されるソフトウェアの概要を紹介しました．しかし，ここで紹介した機能はあくまでも基本的なものであり，実際の機器ではさらに複雑で高度な処理が行われています．新しい技術の市場投入までの期間は，比較的長いと言われる自動車業界でも年々短くなっており，研究・開発期間の短縮が重要な課題となっています．そこに，これまで以上に高度な処理を実装する

ット(MACユニット)」と言います．

積和演算はDSPにとってもっとも重要な機能であり，1秒間に何回の積和演算を実行できるかでDSPの性能を表現する場合があります．例えば，本文で紹介した「Blackfin(BF561)」は，最大3000MMAC/s(1秒間に30億回の積和演算)の性能を持っています．

● ハードウェア・ループ

畳み込み演算のように同じ演算を繰り返す処理の場合，繰り返し回数を制御するための命令に費やされる処理時間を無視できません．例えば，一般的なマイクロプロセッサの場合，演算を1回行うたびにその演算が指定した回数分だけ繰り返されたかどうかを比較する必要があります．つまり，1回の演算と比較の処理時間がそれぞれ1サイクルとすると，演算をN回繰り返すために$2N$サイクルの処理時間がかかってしまいます．

そこで，DSPでは繰り返し回数の制御機構をハードウェアに組み込み，N回の演算のための処理時間がNサイクルですむようにくふうしています．この機構を「ハードウェア・ループ」と呼びます．

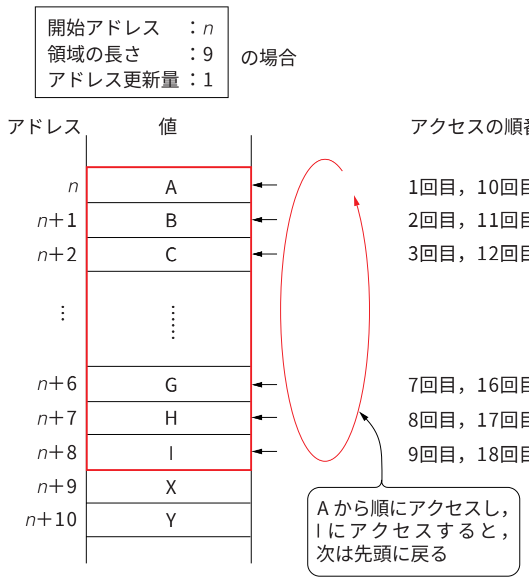
● サーキュラ・アドレッシング(循環アドレッシング)

ディジタル信号処理では，ある有限長のデータ・メモリ領域を連続的に繰り返しアクセスすることがあります．

本文の**図5**の画像フィルタの場合，すべての画素に対して畳み込み演算を行うためにA～Iのマスク係数が保存されたメモリ領域を連続的に繰り返しアクセスします．このとき，A～Iの領域を指定するアドレス・ポインタは順番に更新されていきますが，領域の最後(Iの位置)に来たときはポインタの値を領域の先頭(Aの位置)に戻さなければなりません．

マイクロプロセッサを使ってこの操作を実現するためには，ポインタが更新されるたびに領域の最後に来たかどうかを比較し，来ていれば先頭に戻し，来ていなければそのまま進むという処理が必要となります．これが余計なオーバヘッドとなってしまいます．

そこで，DSPではこれらの処理をハードウェアで行います．あらかじめ領域の先頭と長さを指定しておけば，ポインタの値がつねにその領域を指すように自動的に更新される機構を搭載しています．この機能を「サーキュラ・アドレッシング機能」と呼びます(**図A-2**)．



図A-2　サーキュラ・アドレッシング
開始アドレスと領域の長さをあらかじめ指定すると，アドレスの比較演算なしに，つねにその領域にアクセスできる．

には，プロセッサを利用したソフトウェア技術が必須であり，今後さらに，車載機器におけるプロセッサの重要度は増していくと思われます．

参考・引用*文献

(1) 国土交通省自動車交通局ASVプロジェクトのホームページ．http://www.mlit.go.jp/jidosha/anzen/asv/index.html.
(2) 本田技研工業のホームページ．http://www.honda.co.jp/safety/.
(3) 杉山弘幸；知られざる車載マイコン徹底解剖，Design Wave Magazine，2003年12月号，pp.29-38.
(4) 佐藤道夫，西野信；人と地球にやさしい車を作る組み込み技術，Design Wave Magazine，2004年7月号，pp.28-42.
(5) Analog Devices社のBlackfinに関するのホームページ．http://www.analog.com/processors/Japan/processors/blackfin/.
(6) David Katz，Tomasz Lukasiak，Rick Gentile；Use of Video Technology to Improve Automotive Safety Becomes More Feasible with Blackfin Processors. http://www.analog.com/library/analogDialogue/archives/38-03/auto_safety.html.

すずき・のぶお
アナログ・デバイセズ(株)

<著者プロフィール>
鈴木伸夫．1999年にアナログ・デバイセズに入社．加速度センサやジャイロ・センサなどMEMS製品のサポート担当を経て，DSPのアプリケーション・エンジニアに．現在，Blackfinプロセッサに関するソフトウェア開発環境のトレーニング講師や顧客サポートを担当．